APPLICATION NO. 09/846.410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for CDMA

INVENTOR: Urbain A. von der Embse

Currently amended CLAIMS

APPLICATION NO. 09/846.410

TITLE OF INVENTION: Multiple Data Rate Hybrid Walsh Codes for CDMA

INVENTOR~~S~~: Urbain A. von der Embse

CLAIMS

WHAT IS CLAIMED IS:

Claim 1. (cancelled)

Claim 2. (cancelled)

Claim 3. (cancelled)

Claim 4. (cancelled)

Claim **5.** (currently amended) A method for ~~the~~ implementation of ~~design and implementation of fast encoders and fast decoders for Hybrid Walsh and generalized Hybrid~~ hybrid Walsh complex orthogonal codes for CDMA, ~~channelization codes for multiple data rate users over~~ said method comprising the steps: ~~a frequency band with properties~~

generating N Walsh codes W(c) with code index c=0,1,2,. . .,N-1, each with N chips where N is a power of 2,

generating said N hybrid Walsh codes $\tilde{W}(c)$ by re-ordering said Walsh codes defined by equations

for c = 0, $\tilde{W}(c) = W(0) + jW(0)$

for c = 1,2,…,N/2-1, $\tilde{W}(c) = W(2c) + jW(2c-1)$

for c = N/2, $\tilde{W}(c) = W(N-1) + jW(N-1))$

for c = N/2+1,…,N-1, $\tilde{W}(c) = W(2N-2c-1) + jW(2N-2c)$

wherein $j=\sqrt{-1}$,

wherein said hybrid Walsh codes are generated by reading code chip values from said Walsh code memory in a digital signal processor and writing to said hybrid Walsh memories using addresses specified by said re-orderings of said Walsh codes and,

applying said hybrid Walsh codes in the encoder and in the decoder by replacing existing said Walsh real codes with said hybrid Walsh complex codes using the same code vector indexing.

~~Hybrid Walsh inphase (real axis) codes and quadrature (imaginary axis) codes are defined by lexicographic reordering permutations of the Walsh code~~

~~Hybrid Walsh codes have a 1-to-1 sequency-frequency correspondence with the DFT codes and have a 1-to-1 even-cosine and odd-sine correspondences with the DFT codes~~

~~Hybrid Walsh codes take values {1+j, -1+j, -1-j, 1-j} or equivalently take values {1, j, -1, -j} with a (-45) rotation of axes and a renormalization~~

~~generalized Hybrid Walsh codes can be constructed for a wide range of code lengths by combining Hybrid Walsh with DFT (discrete Fourier transform), Hadamard and other orthogonal codes, and quasi-orthogonal PN codes using tensor product, direct product, and functional combining~~

~~fast encoding and fast decoding implementation algorithms are defined~~

~~algorithms are defined to map multiple data rate user data symbols onto the code input data symbol vector for fast encoding and the inverses of these algorithms are defined for recovery of the data symbols with fast decoding~~

~~encoders perform complex multiply encoding of complex data to replace the current Walsh real multiply encoding of inphase and quadrature data~~

~~decoders perform complex conjugate transpose multiply decoding of complex data to replace the current Walsh real multiply decoding of inphase and quadrature data~~

Claim **6.** (currently amended) A method for ~~the~~ implementation of ~~design and implementation of encoders and decoders for complex orthogonal CDMA and generalized~~ hybrid Walsh codes for CDMA as described in claim **5**, further comprising the steps: ~~complex orthogonal CDMA channelization codes for multiple data rate users over a frequency band with properties~~

using tensor products also called Kronecker products to construct a second code,

wherein an example 24 chip tensor product code is constructed from a 8 chip hybrid Walsh code and a 3 chip discrete Fourier transform DFT code,

said 24 chip code is defined by a 24 row by 24 column code matrix C_{24} wherein row vectors are code vectors and column elements are code chips,

said 8 chip hybrid Walsh code is defined by a 8 row by 8 column code matrix $\widetilde{W}_8$,

said 3 chip DFT code is defined by a 3 row by 3 column code matrix E_3,

said C_{24} is constructed by tensor product of said $\widetilde{W}_8$ with said E_3 defined by equation

$$C_{24} = \widetilde{W}_8 \otimes E_3$$

wherein symbol "$\otimes$" is a tensor product operation,

row u+1 and column n+1 matrix element $C_{24}(u+1,n+1)$ of said C_{24} is defined by equation

$$C_{24}(u+1,n+1) = \tilde{W}_8(u_0+1,n_0+1)\ E_3(u_1+1,n_1+1)$$

wherein

$$u = u_0 + 8u_1$$
$$u = 0,1,\ldots,23$$
$$n = n_0 + 8n_1$$
$$n = 0,1,\ldots,23$$

wherein u,n are code and chip indices for said codes C_{24} and u_0,n_0 are code and chip indices for said code $\tilde{W}_8$ and u_1,n_1 are code and chip indices for said code E_3,

wherein said encoder and said decoder for CDMA communications have memories assigned to said C_{24}, $\tilde{W}_8$, E_3 codes,

said C_{24} codes are generated by reading code chip values from said $\tilde{W}_8$ memory and said E_3 memory and combining using said equations to yield said chip values for said C_{24} codes and stored in said memory C_{24},

said C_{24} codes are read from said memory and implemented in said encoder and said decoder,

using direct products to construct a second code.

wherein an example 11 chip direct product code is constructed from said 8 chip hybrid Walsh code and said 3 chip DFT code,

said 11 chip code is defined by the 11 row by 11 column code matrix C_{11},

said C_{11} is constructed by direct product of said $\tilde{W}_8$ with said E_3 defined by equation

$$C_{11} = \tilde{W}_8 \oplus E_3$$

wherein symbol "$\oplus$" is a direct product operation,

row u+1 and column n+1 matrix element $C_{11}(u+1,m+1)$ of said C_{11} is

defined by equation

$$
\begin{aligned}
C_{11}(u+1,n+1) &= \widetilde{W}_8(u_0+1,n_0+1) && \text{for } u=u_0,\ n=n_0, \\
&= E_3(u_1+1,n_1+1) && \text{for } u=8+u_1,\ n=8+n_1, \\
&= 0 && \text{otherwise,}
\end{aligned}
$$

wherein said encoder and said decoder for CDMA communications have memories assigned to said C_{11}, $\widetilde{W}_8$, E_3 codes,

said C_{11} codes are generated by reading code chip values from said $\widetilde{W}_8$ memory and said E_3 memory and combined using said equations to yield said chip values for said C_{11} codes and stored in said C_{11} memory,

said C_{11} codes are read from memory and implemented in said encoder and said decoder,

using functional combining to construct a second code,

wherein an example 11 chip functional combined $\hat{C}_{11}$ code is constructed from said C_{11} codes by using codes to fill the two null subspaces of said C_{11}.

wherein said $\hat{C}_{11}$ codes are read from memory and implemented in Said encoder and said decoder and,

using a combinations of tensor products, direct products, and functional combining to construct said codes which are read from memory and implemented in said encoder and said decoder.

~~complex codes inphase (real axis) codes and quadrature (imaginary axis) codes are defined by reordering permutations of the real Walsh codes~~

~~generalized complex codes can be constructed for a wide range of code lengths by combining the complex codes with DFT (discrete Fourier transform), Hybrid Walsh, Hadamard and other~~

~~orthogonal codes, and quasi-orthogonal PN codes using tensor product, direct product, and functional combining~~

~~fast encoding and fast decoding implementation algorithms are defined~~

~~algorithms are defined to map multiple data rate user data symbols onto the code input data symbol vector for fast encoding and the inverses of these algorithms are defined for recovery of the data symbols with fast decoding~~

~~encoders perform complex multiply encoding of complex data to replace the current Walsh real multiply encoding of inphase and quadrature data~~

~~decoders perform complex conjugate transpose multiply decoding of complex data to replace the current Walsh real multiply decoding of inphase and quadrature data~~

Claim **7.** (currently amended) A method for implementation of Hybrid Walsh codes for CDMA, further comprising the steps:

said encoder operates as a block encoder,

encoding blocks of received N data symbols with said N hybrid Walsh codes and summing to yield N chips for each block at the output chip rate 1/T chips per second,

wherein said encoder accepts up to M users per block for N=2*M,

said users have data rates from the menu 1/NT,2/NT,. . .,2/T respectively corresponding to 1,2,. . . ,N/2 said user data symbols over said block,

user data symbols over said block are arranged in packets with each packet containing said user data symbols for said block,

said encoder accepts packets from each user and writes them to memory "A" for each block,

binary address index $d=d_0+2d_1+4d_2+...+(N/2)d_{M-1} = 0,1,...,N-1$ is used for addressing of said data symbols stored in "A" wherein binary coefficients $d_0,d_1,...,d_{M-1}$ take values 0,1,

said binary address index can be independently mapped onto said data symbol addresses of "A" to provide additional flexibility in assigning users to hybrid Walsh vectors,

said data symbol address is partitioned into M overlapping algebraic index fields d_{M-1}, $d_{M-2}d_{M-1}$, . . ., $d_1d_2\cdots d_{M-2}d_{M-1}$, $d_0d_1d_2\cdots d_{M-2}d_{M-1}$, with each field indexed over the allowable number 2,4,. . .,N/2,N of said data rate users at symbol rates 1/2T,1/4T,. . .,2/NT,1/NT respectively,

assign said users with like data symbol rates to the M groups u_0, u_1, . . ., u_{M-2}, u_{M-1}, of users with the respective symbol rates 1/2T,1/4T,. . .,2/NT,1/NT,

assign said data symbol indices in said index field d_{M-1} to said users in said group u_0, assign said data symbol indices in said index ield $d_{M-2}d_{M-1}$ to said users in said group u_1, et al and finally assign said data symbol indices in said index field $d_0d_1d_2\cdots d_{M-2}d_{M-1}$ to said users in said group u_{M-1},

use said mapping and assignments to specify said write addresses of said user data symbols onto said input code vector stored in said memory "A" and,

said input vector in said "A" is encoded in said encoder and processed for transmission.

Claim **8**. (currently amended) Wherein said hybrid Walsh codes in claim **5** have a fast encoding implementation algorithm, comprising the steps:

said fast implementation algorithm in encoder uses said memory "A" for input and to support pass 1, memories "B","C" to support passes 2,. . . ,M and re-ordering pass, and memory "D". for output,

write input data symbol vector $Z(d_0, d_1, . . ., d_{M-2}, d_{M-1})$ to said "A" wherein said $(d_0, d_1, . . ., d_{M-2}, d_{M-1})$ is said binary addressing index after said mapping of said data vector onto said "A",

pass 1 reads from said "A", performs pass 1, and writes the output to said "B",

pass 1 multiplies said Z by the kernel $[(-1)^{\wedge}dr_0 n_{M-1}+j(-1)^{\wedge}di_0 n_{M-1}]$ and sums over $dr_0, di_0=0,1$ to yield the partially encoded symbol set $Z(n_{M-1}, d_1, . . ., d_{M-2}, d_{M-1})$ where $dr_0=cr(d_0)$ and $cr(d)$ is the real axis Walsh code for d, $di_0=ci(d_0)$ where $ci(d)$ is the imaginary axis Walsh code for d, and n_{M-1} is a binary code chip coefficient in said code chip indexing $n = n_0+2n_1+. . .+(N/4)n_{M-2}+(N/2)n_{M-1}$,

write said output symbol set $Z(n_{M-1}, d_1, . . ., d_{M-2}, d_{M-1})$ to said "B" wherein said address index n_{M-1} replaces said index d_0,

pass 2 reads from said "B", performs pass 2, and writes the output to said "C",

pass 3 reads from said "C", performs pass 3, and writes the output to said "B",

subsequent passes alternate in read/write from/to said "B" and write/read to/from said "C",

implement passes m=2,3,. . .,M-1 of said fast encoding algorithm by multiplying

$Z(n_{M-1}, n_{M-2} . . ., n_{M-m+1}, d_{m-1}, . . ., d_{M-2}, d_{M-1})$ by the kernel

$[(-1)^{\wedge}dr_{m-1}(n_{M-m} + n_{M-m+1})+j(-1)^{\wedge}di_{m-1}(n_{M-m} + n_{M-m+1})]$ and summing over $dr_{m-1}, di_{m-1}=0,1$ to yield the partially encoded symbol set $Z(n_{M-1}, n_{M-1}, n_{M-2} . . ., n_{M-m}, d_m, . . ., d_{M-2}, d_{M-1})$,

implement pass M of said fast encoding algorithm by by multiplying $Z(n_{M-1}, n_{M-2} . . ., n_2, n_1, d_{M-1})$ by the kernel

$[(-1)^{\wedge}dr_{M-1}(n_0 + n_1)+j(-1)^{\wedge} di_{M-1}(n_0 + n_1)]$ and summing over $dr_{M-1}, di_{M-1}=0,1$ to yield the encoded symbol set

$Z(n_{M-1}, n_{M-1}, n_{M-2} . . . ,n_2, n_1, n_0)$, reorder said encoded symbol set in memory in the ordered output format $Z(n_0, n_1, . . . ,n_{M-2}, n_{M-1})$ and store in said "D" and, said encoder in said transmitter reads said encoded symbol vector in said "D" and overlays said vector with long and short PN codes to generate N chips of said hybrid Walsh encoded data symbol vector for subsequent processing and transmission.

Claim **9.** (currently amended) Wherein said hybrid Walsh codes in claim **5** have a fast decoding implementation algorithm, comprising the steps:

said decoder in said receiver strips off said PN codes from said received N chip encoded data symbol vector and outputs said received hybrid Walsh encoded chip vector $Z(n_0, n_1, . . . ,n_{M-2}, n_{M-1})$ for implementation of said fast decoding algorithm,

said fast implementation algorithm in said decoder uses memory "E" for input and to support pass 1, memories "F","G" to support passes 2,3, . . . ,M and re-ordering pass, and memory "H" for output,

write said $Z(n_0, n_1, . . . ,n_{M-2}, n_{M-1})$ to said "E" wherein $(n_0, n_1, . . . ,n_{M-2}, n_{M-1})$ is the binary address,

pass 1 reads from said "E", performs pass 1, and writes the output to said "F",

implement pass 1 of said fast decoding algorithm by multiplying said $Z(n_0, n_1, . . . ,n_{M-2}, n_{M-1})$ by the kernel [(-1)^n_0dr_{M-1}+j(-1)^n_0di_{M-1}] and summing over n_0=0,1 to yield the partially decoded symbol set

$Z(d_{M-1}, n_1, . . . ,n_{M-2}, n_{M-1})$,

write said output symbol set $Z(d_{M-1}, n_1, . . . ,n_{M-2}, n_{M-1})$ to said "F" wherein address index d_{M-1} replaces index n_0,

pass 2 reads from said "F", performs pass 2, and writes the output to said "G",

pass 3 reads from said "G", performs pass 3, and writes the output to said "F",

subsequent passes alternate in read/write from/to said "F" and write/read to/from said "G",

implement passes m=2,3,. . .,M-1 of said fast decoding algorithm by multiplying $Z(d_{M-1}, d_{M-2}, \ldots, d_{M-m+1}, n_{m-1}, \ldots, n_{M-2}, n_{M-1})$ by the kernel $[(-1)\hat{}n_{m-1}(dr_{M-m} + dr_{M-m+1}) + j(-1)\hat{}n_{m-1}(di_{M-m} + di_{M-m+1})]$ and summing over $n_{m-1}=0,1$ to yield the partially decoded symbol set $Z(d_{M-1}, d_{M-1}, d_{M-2}, \ldots, d_{M-m}, n_m, \ldots, n_{M-2}, n_{M-1})$,

implement pass M of said fast decoding algorithm by by multiplying $Z(d_{M-1}, d_{M-2}, \ldots, d_2, d_1, n_{M-1})$ by the kernel $[(-1)\hat{}n_{M-1}(dr_0 + dr_1) + j(-1)\hat{}n_{M-1}(di_0 + di_1)]$ and summing over $n_{M-1}=0,1$ and rescaling by dividing by 2N to yield the decoded symbol set $Z(d_{M-1}, d_{M-1}, d_{M-2}, \ldots, d_2, d_1, d_0)$,

reorder said decoded symbol set in the ordered output format $Z(d_0, d_1, \ldots, d_{M-2}, d_{M-1})$ and store in said "H" and,

said decoder in said receiver reads said decoded symbol vector in "D" , re-orders the read data symbols to remove said mapping onto said "A", and performs subsequent receive signal processing to recover the information from the data symbols..